第 38 卷第 3 期
录用定稿
计算机应用研究
Application Research of Computers
Vol. 38 No. 3
Accepted Paper

基于移动边缘计算的车联网缓存策略研究 *

刘可欣，陈桂芬†

(长春理工大学 电子信息工程学院，长春 130022)

摘　要：针对车联网中数据流量爆炸式增长而引起的业务响应时延过高的问题，提出了一种基于移动边缘计算的蚁群模拟退火算法缓存策略(ACSAM)。首先，在基于 5G 的“车-边-云”协同系统架构下，以最小化内容下载时延为目标，建立了通信计算模型；其次，采用蚁群算法构造了使内容下载时延最小的局部最优解；最后，使用模拟退火算法对局部最小下载时延进行扰动，并以一定概率接受新解，从而得到全局最小下载时延，即保证了内容被预缓存在最佳的位置。仿真结果表明，在车边云协同架构下，ACSAM 缓存策略可显著减少传输冗余，降低下载时延。

关键词：车联网；移动边缘计算；缓存策略

中图分类号：TP391　　doi: 10.19734/j.issn.1001-3695.2019.12.0700

Research on caching strategy of Internet of vehicles based on mobile edge computing

Liu Kexin, Chen Guifen†

(*School of Electronics & Information Engineering*, *Changchun University of Science & Technology*, *Changchun* 130022, *China*)

Abstract: To solve the problem of high service response delay caused by explosive growth of data traffic in the Internet of vehicles, this paper proposed an Ant Colony Simulated Annealing algorithm cache strategy based on Mobile edge computing(ACSAM). Firstly, under the 5G based " vehicle-edge-cloud " collaborative system architecture, established a communication computing model to minimize the content download delay; secondly, used the Ant Colony Optimization to construct a local optimal solution to minimize the content download delay; finally, used the search ability of the Simulated Annealing algorithm to disturb the local optimal solution and accept the new solution with a certain probability, thereby obtained the global optimal solution and improved cache hit rate. Simulation results show that under the vehicle-edge-cloud collaborative architecture, the ACSAM cache strategy can significantly reduce transmission redundancy and reduce download latency.

Key words: Internet of vehicles; mobile edge computing; caching strategy

0　引言

车联网作为 5G 的主要应用场景之一，是 2015 年由国务院发布的“中国制造 2025”战略的焦点[1]。随着车联网用户对各种服务需求的增加，使得网络中的业务流量大幅度增长，通过观察移动流量[2,3]可以发现，大多数的流量请求是由少数内容产生的，即流行内容被重复请求的概率大。随着 2021 年的到来，每月的全球移动流量将有极大的可能性超出 35 艾字节。为了解决数据流量大幅度增长所导致的响应时延过高的问题，移动边缘计算(mobile edge computing，MEC)[4]以其特有的优势被引入到了 5G 超密集部署的车联网中。通过在流量非高峰时段，将流行内容预先缓存在部署于路侧单元或基站的 MEC 服务器，可以有效减少网络流量损耗和内容传输冗余，同时降低业务响应时延，提供用户高质量的服务体验。

MEC 由移动云计算演化而来，首次出现是在 2013 年。欧洲电信标准协会(ETSI)于 2014 年正式将 MEC 定义为一种既可以提供信息技术服务环境又具有云计算能力的体系架构。目前，作为 5G 的关键技术之一，MEC 可以动态地获取无线接入网络内的实时信息，促进缓存技术朝着更加智能化的方向发展。通过将 MEC 和缓存进行整合，可实现服务响应的高效性，为此已有大量学者对其开展了研究。文献[5]中，作者在多个 cloudlet 协作的分布式系统框架下，制定了内容分配优化模型，提出了以状态为前提的内容分发算法，显著提高了内容缓存命中率。文献[6]提出了一个联合的协作式缓存和处理框架，该框架支持 MEC 网络中的自适应比特率(ABR)视频流，可有效降低内容交付延迟。文献[7]提出了一种基于深度学习的缓存方案，通过准确地预测内容流行度并确定将哪些内容存储在基站，保证了缓存决策的有效性。文献[8]提出了一种新的基于移动感知概率边缘缓存方案(mobility-aware probabilistic edge caching strategy，MAP)，该方案充分利用了无线接入网上下文数据，可最大限度提高缓存命中率并最小化内容检索延迟。文献[9]将车辆层和路侧单元层缓存相结合，提出了基于云的双层缓存放置策略(double layer cache placement strategy，DLC)，可实现内容在最佳节点的缓存，保证任务快速响应。

虽然研究人员已对缓存方案进行了大量研究，但大多没有考虑移动性对缓存过程造成的影响，同时也没有考虑移动终端、MEC 及云的相互协作。因此，本文以车联网为背景，在“车—边—云”协同的系统架构下，建立了基于时延的通信模型，并提出了一种基于 MEC 的蚁群模拟退火算法缓存策略(ant colony and simulated annealing algorithm cache strategy based on Mobile edge computing，ACSAM)。仿真结果表明，在“车-边-云”协同架构下，该缓存策略可有效降低业务响应延迟。

1　系统架构及场景分析

1.1　系统架构

为了保证车辆与其他实体信息交换(V2X)的实时性，5G

收稿日期：2019-12-20；**修回日期：**2020-04-08　**基金项目：**吉林省科技厅资助项目(20190302103GX)

作者简介：刘可欣(1995-)，女，吉林德惠人，硕士研究生，主要研究方向为边缘计算；陈桂芬(1964-)，女(通信作者)，吉林九台人，教授，博士，主要研究方向为无线通信系统理论与技术(chenguif@163.com).

使能技术与车联网结合是大势所趋，具体的协同系统架构如图 1 所示。

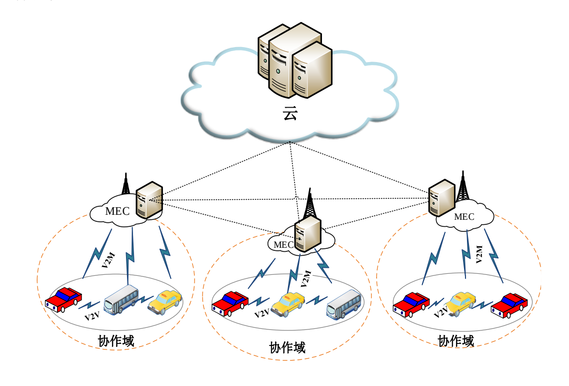


图 1　基于 5G 的“车-边-云”协同系统架构

Fig. 1　5G-based "vehicle-edge-cloud" collaborative system architecture

在该架构中融入了多种新兴技术，其中，软件定义网络(Software Defined Network, SDN)可以对多个 MEC 服务器进行控制，并且统一配置 MEC 与云平台的资源；网络功能虚拟化(Network Function Virtualization, NFV)以整合网络设备类型为目标，可以管理多种不同架构的设备[10]，使网络的可扩展性及灵活性得到改善；网络切片能够保证车联网业务运行的独立性，并实现相应的功能定制；MEC 可以将第三方应用及业务部署在网络边缘，实现基站与业务的深度融合，将其与车、云协同可减少数据传输冗余，获得比“边-云”协同更低的数据投递时延。

1.2　场景分析

在 5G 超密集部署的车联网场景中，车辆移动带来了通信的不稳定，同时也一定程度上增加了车辆通信的机会。在车-边-云架构中包括一个中央云、多个 MEC 服务器及车辆，可进行协作式缓存并且允许车与车之间直接通信(V2V)。MEC 服务器与车辆都具有一定的缓存能力，当有车辆发出某一请求时，主要通过以下四种方式获取请求内容:

a) 如果车辆恰好存储该内容，那么就可以直接从源车辆获取，此时响应时延为零;

b) 向一定通信范围内的 V2V 车辆发出请求，若 V2V 车辆中已存储该车辆所请求的内容，则将该内容通过 V2V 链路直接传送给该车辆;

c) 向在路侧单元等位置部署的 MEC 服务器发出请求，若在本地的 MEC 服务器未缓存该请求内容，则通过其他协作的 MEC 服务器获取，之后，再经由本地 MEC 服务器传送给请求车辆;

d) 由中央云通过核心网将请求内容传到本地 MEC 服务器，再经本地 MEC 服务器将该内容传输给请求车辆。

由于请求内容从核心网传输到发出请求的车辆会增加成本和时延，通过将流行内容预先缓存在车辆和 MEC 服务器中，可大大减少下载时延，节省带宽资源。因此，本文在“车-边-云”协同架构下，以最小化内容的下载时延为目标，提出了 ACSAM 缓存策略。

2　通信计算模型

假设一个缓存域内有 M 个 MEC 服务器和 N 个车辆，用集合表示为 $V=E\cup U=\{v_1,v_2,\ldots,v_{M+N}\}$，其中 $E=\{e_1,e_2,\ldots,e_M\}$ 表示 MEC 设备的集合，$U=\{u_1,u_2,\ldots,u_N\}$ 表示车辆的集合。$N(i)\subset V$ 是由能够与车辆进行通信的 V2V 车辆和能够为车辆提供服务的 MEC 服务器所组成的集合。$u(i)=N(i)\cap U$ 表示车辆 u_i 的 V2V 邻居集合，$m(i)=N(i)\cap E$ 表示为车辆 u_i 提供服务的 MEC 集合。假设数据中心将 L 个不同的内容文件 $O=\{o_1,o_2,\ldots,o_L\}$ 提供给车联网，每个内容的长度归一化为 s。

定义内容流行度服从 Zipf 分布，则内容 o_f 被请求的概率为

$$p_f=\frac{f^{-\theta}}{\sum_j^L j^{-\theta}} \tag{1}$$

其中，θ 为 Zipf 分布的影响因子。假设车辆发出的请求服从期望为 θ_i 的泊松分布，则在 Δt 内请求内容 o_f 的概率为 $p_{if}=p_f\cdot\theta_i$。

定义内容缓存矩阵 $X=\{x_{m,f}\mid e_m\in E,o_f\in O\}$ 和 $Y=\{y_{i,f}\mid u_i\in U,o_f\in O\}$，具体如下：

1) 在 MEC 缓存

$$x_{m,f}=\begin{cases}1, & \text{内容}o_f\text{预缓存在MEC服务器}\\0, & \text{内容}o_f\text{预缓存在V2V车辆}\end{cases} \tag{2}$$

2) 在车辆缓存

$$y_{i,f}=\begin{cases}1, & \text{内容}o_f\text{预缓存在源车辆}u_i\\0, & \text{内容}o_f\text{预缓存在V2V车、MEC 服务器}\end{cases} \tag{3}$$

定义系统缓存命中率为在 MEC 和车辆获取的内容请求数与总请求数之比，如式(4)所示。

$$R=\frac{Q}{Z} \tag{4}$$

其中，Z 表示缓存域内车辆发出的总请求数，Q 表示在车辆和 MEC 缓存命中的请求数。

针对车的移动性，这里将 MEC 服务器和车辆协同缓存问题转换为最小化下载时延问题。

1) V2V 通信时延

设 R_{V2V} 为车与车间可进行通信的最大距离，不同 V2V 对的通信是互不干扰的。车辆 u_i 和车辆 u_j $(u_i\in U,u_j\in U(i))$ 的信息交换时间 $\gamma_{ij}(for\forall u_i\in U,u_j\in U(i))$ 服从期望值为 $1/\lambda_{ij}(\lambda_{ij}>0)$ 的指数分布，其概率密度可表示为

$$f_{ij}(t)=\lambda_{ij}\cdot e^{-\lambda_{ij}t},t\ge 0 \tag{5}$$

其中，$1/\lambda_{ij}$ 表示 u_i 和 u_j 的平均通信时长，λ_{ij} 越大，表示车辆移动强度越大。若在 γ_{ij} 内，两车间传输的数据比特数不小于请求内容 o_f 的大小，便可以保证内容传输的连续性。

定义 u_i 通过 V2V 通信链路成功获取到请求内容 o_f 的概率为 $P_{i,f}^{v2v}$，那么:

$$P_{i,f}^{v2v}=1-\prod_{u_j\in U(i)}(1-y_{j,f}\cdot e^{-y_{j,f}\cdot\lambda_{ij}\cdot t_{i,u_j}}) \tag{6}$$

设定车与车采用 DSRC 通信方式中的 IEEE802.11p 协议[11]在独立同分布信道中进行通信，路径损耗[12]可以定义为

$$L_{v2v}^{dB}=63.3+17.7\log_{10}(d_{i,u_j}) \tag{7}$$

其中，d_{i,u_j} 表示车辆 u_i 与 u_j 之间的通信距离，式(7)的单位为 dB，转换为数值形式为

$$L_{v2v}=10^{-\frac{L_{v2v}^{dB}}{10}} \tag{8}$$

因此，u_i 从 V2V 车辆 u_j 获取内容的下载速度 w_{i,u_j} 为

$$w_{i,u_j}=B_{i,u_j}\cdot\log_2\left(1+\frac{p_v L_{v2v}\kappa}{n_0}\right) \tag{9}$$

其中，P_v 表示车辆的发射功率，n_0 表示高斯白噪声功率，B_{i,u_j} 是车与车通信链路的带宽，κ 表示路径损耗常数。

则 u_i 从 V2V 车辆 u_j 获取内容的下载时延 t_{i,u_j} 为

$$t_{i,u_j}=s/w_{i,u_j} \tag{10}$$

2) V2M 通信时延

设定车辆和 MEC(RSU)间的通信链路为频率平坦型块衰落的瑞利信道，则 u_i 从 MEC 服务器获取内容的下载速度 w_{i,e_j} 为

$$w_{i,e_j}=B_{i,e_j}\cdot\log_2\left(1+\frac{p_m d_{i,e_j}^{-\varepsilon}\kappa}{n_0}\right) \tag{11}$$

其中，P_m 表示 MEC 发射功率，d_{i,e_j} 表示 MEC 服务器与接收车辆间的距离，κ 表示路径损耗常数，B_{i,e_j} 是 MEC 到车的通信链路带宽，ε 为路径损耗指数。

则 u_i 从 MEC 服务器获取内容的下载时延 t_{i,e_j} 为

$$t_{i,e_j}=s/w_{i,e_j} \tag{12}$$

一般而言，内容从云平台到本地 MEC、MEC 到 MEC 的传输时延均可视为常量，这里分别用 t_0 、t_1 表示，则车辆 u_i 以 MEC 服务器为媒介获取某一内容的下载时延为

$$t_m=\alpha_{m(i),\ f}(t_{i,e_j}+t_1)+x_{m(i),f}\cdot t_{i,e_j}+\beta_f(t_0+t_{i,e_j}) \tag{13}$$

$$s.t.\quad x_{m,f},y_{i,f}\in\{0,1\},\forall e_m\in E,u_i\in U,o_f\in O \tag{13a}$$

$$\alpha_{m(i)f}=(1-x_{m(i),f})\cdot(1-\prod\nolimits_{m=1}^{M}(1-x_{m,f})),\forall o_f\in O \tag{13b}$$

$$\beta_f=\prod\nolimits_{m=1}^{M}(1-x_{m,f}),\forall o_f\in O \tag{13c}$$

其中，式(13-a)表示 x ， y 取值，式(13-b)表示仅当 o_f 被缓存在除本地 MEC 外的其他协作 MEC 中时，$\alpha_{m(i),f}$ 的值为 1；式(13-c)表示内容 o_f 仅在云服务器缓存时， β_f 的值为 1。

综上，请求内容被各车辆获取的总下载时延 $t(L)$ 为

$$t(L)=\sum\nolimits_{i=1}^{N}\sum\nolimits_{f=1}^{L}p_{if}(1-y_{i,f})\left\{p_{i,f}^{v2v}\cdot t_{i,u_j}+(1-p_{i,f}^{v2v})\cdot t_m\right\} \tag{14}$$

为保证缓存位置最佳，响应时延最小，MEC 与车间的协作缓存优化问题可以表示为

$$\min\{t(L)\} \tag{15}$$

3　ACSAM 内容缓存策略

蚁群算法(ant colony optimization，ACO)是受蚂蚁觅食过程中产生的一系列行为的启迪，以信息素浓度为依据选择行进路线的一种算法。当某一路径上有较高的信息素浓度存留时，该路径就有极大的可能性被选中。基本的 ACO 算法具有并行性等其他算法无法比拟的优势，但在求解如上述所示的缓存优化这类 NP-hard 问题时，存在明显的不足，即收敛速度较缓慢，难以跳出某一局部区域的最优解。

模拟退火算法(simulated annealing algorithm，SA)是一种随机优化算法，该算法只与初始温度、最小温度和退火次数有关，其优点是能够有效解决 NP-hard 问题，并在解空间找到符合条件的全局最佳结果，即能够避免问题的解在一定区域内最优，缺点是随机性较重。

考虑到蚁群和模拟退火算法自身的优缺点，将这两种算法相结合既可以提高退火的速度，又可以避免陷入局部最优。因此，本文提出了 ACSAM 内容缓存策略，通过将流行内容预先缓存在 MEC 服务器和车辆中，可显著降低传输造成的时间及成本损耗，保证响应速度。

3.1　启发函数设置

假设车从某一 MEC 服务器或车辆下载内容的时延为

$$t_{i,v_j}=\frac{O_totallength}{w_{i,v_j}} \tag{16}$$

其中，$O_totallength$ 为车从 V_j 下载的内容总长度，w_{i,v_j} 为下载速度。

定义启发式函数 $\eta_j(t)$ 为

$$\eta_j(t)=\left(1\text{-}\frac{t_{i,v_j}-t_{ave}}{t_{i,v_j}+t_{ave}}\right)^2 \tag{17}$$

其中，t_{ave} 表示在当前解最佳的情况下从各个设备下载内容的平均时延。 $\eta_j(t)$ 越大， V_j 被选中概率越大。

3.2　状态转移概率

利用蚂蚁搜寻路径，每一步的行进方向都由路径上存留的信息素浓度 $\tau_j(t)$ 判定，并且每步转移的可能性都按照式(18)进行计算，重复该过程直到到达(或找不到)目的地退出。

$$p_j(t)=\begin{cases}\dfrac{[\tau_j(t)]^\alpha.[\eta_j(t)]^\beta}{\sum\limits_{s\in V_k}[\tau_s(t)]^\alpha.[\eta_s]^\beta}, & 当j\in V_k时\\ 0, & 其他\end{cases} \tag{18}$$

其中，$tabu_k$ 为蚂蚁走过的地点集合，$V_k=\{1,2,\cdots,\mathrm{M}+\mathrm{N}\}-tabu_k$ 表示蚂蚁前进过程中即将选取的车辆和 MEC 设备的集合，α 、 β 分别表示信息素和启发式因子的相对重要程度。

3.3　信息素更新规则

a) 当某一蚁群执行任务结束时，对该蚁群所访问过的节点的信息素浓度进行更新(局部信息素更新)。

$$\tau_j(t+n)=(1-\rho)\cdot\tau_j(t)+\Delta\tau_j(t) \tag{19}$$

其中， $\rho(0<\rho<1)$ 表示信息素的蒸发系数， $\Delta\tau_j(t)=1/t_j$ ， t_j 表示某次迭代时发出请求的车辆从 V_j 上下载内容所用的时间。

b) 当所有蚁群都完成任务时，更新全局信息素。

$$\tau_j(t+n)=(1-\rho)\cdot\tau_j(t)+\Delta\tau_j^{'}(t) \tag{20}$$

其中 $\Delta\tau_j^{'}(t)=1/t_{bestj}$ ， t_{bestj} 表示在取得全局最优解后，车从 V_j 完成内容下载的时间。

3.4　Metropolis 准则

比较当前温度 T 与前一时刻温度下的解(指用 ACO 优化得到的解)，利用 SA，将当前区域内的最优解通过置换规则扰动，即任意挑选两个内容文件，若二者交换下载地点后响应时间缩短，则接受新解。在以 Metropolis 准则为基本依据的前提下，新解将以式(20)的概率被接受。

$$p=\begin{cases}\exp(-\dfrac{t(L)-\overline{t(L)}}{T}) & t(L)-\overline{t(L)}>0\\ 1 & 其他\end{cases} \tag{21}$$

其中，$t(L)$ 表示在当前温度 T 下，交换地点后获取内容的总下载时延； $\overline{t(L)}$ 表示通过蚁群算法所得到的局部最短下载时间。

3.5　ACSAM 策略的基本流程

a) 初始化参数，包括迭代次数、信息素浓度、初始温度和终止温度等。

b) 利用蚂蚁搜寻路径，并按照式(18)构造候选解。

c) 按照所有内容下载完成时间最短的原则计算出局部最优解，并根据式(19)更新局部信息素。

d) 利用 SA，将当前区域内的最优解通过置换规则扰动以构造新解，并通过 Metropolis 规则决定新解被接受的概率。

e) 判断当前 T 下得到的局部最优解是否符合抽样稳定准则(即在同一温度下，经过连续干扰，局部最优解都固定)，满足继续，否则返回步骤 d) 。

f) 根据式(20)更新全局信息素浓度。

g) 当回火次数达到或当前温度低于最小回火温度，则输出最优解，否则回到步骤 b) 。

3.6　ACSAM 算法伪代码

```
输入：参与的车辆和 MEC 设备。
输出：保证内容缓存在最佳位置的最小下载时延。
a) 初始化参数，包括 α， β， ρ， T_min， T_0 等。
b) While  T > T_min
      for each ant
         随机选择一个初始位置 j；
         for j=1 to M+N
            按照式(18)的概率选择下一缓存位置；
         end for
      end for
      计算第 k 次迭代所用时间；
      for each edge
         按照式(19)更新局部信息素值；
         根据模拟退火的置换准则构造新解，按照式(21)接受新解；
         按照式(20)更新全局信息素 ；
      end for
   end while
c) 输出最小内容下载时延。
```

4　仿真实验

为了验证在“车-边-云”协同架构中 ACSAM 缓存策略

在降低时延方面的优越性，将其与 ACO、MAP[8]、DLC[9]及随机缓存策略(random-based caching strategy ，RC)[13]的缓存性能进行了对比。

4.1　仿真环境

在仿真实验中，设置发布的内容数量为 1000 个，车与车间最大通信距离为 150m，内容流行度服从参数为 0.56 的 Zipf 分布[14]，每个车辆发出的请求服从期望为 0.3 的泊松分布[15]。具体参数设置如表 1 所示。

表 1　相关仿真参数

Tab. 1　Relevant simulation parameters

参数	取值
内容数量	1000
V2V最大通信范围/m	150
MEC(RSU)覆盖半径/m	350
MEC设备发射功率/W	40
车载发射功率/W	1.3
高斯白噪声/W	3×10^{-13}
总带宽/MHz	30
路径损耗常数	10^{-2}
路径损耗指数	4
Zipf分布影响因子	0.56
泊松分布参数	0.3
指数分布参数	Γ(2,0.1)
α	0.3
β	1
ρ	0.4
T_0	1000
$T_{\min}$	10

4.2　仿真结果分析

图 2 仿真对比了在迭代次数影响下 ACSAM、ACO 的时延性能。假定 Zipf 分布的影响因子为 0.56，协作缓存域内的 MEC 和车辆数分别为 10 和 20，服务内容数量为 1000。由图 2 可知，两者的下载时延都随着迭代次数的增加而降低，并且相比于 ACO，ACSAM 的收敛速度更快，下载时延更低，从中可看出 ACSAM 在降低时延方面有着明显优势。

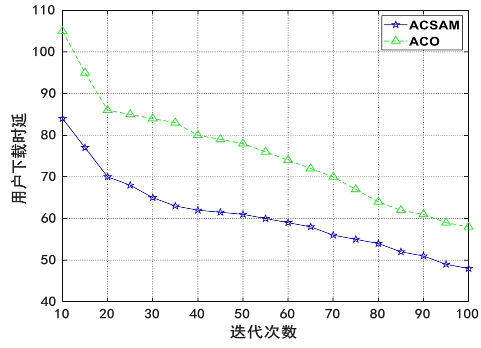


图 2　迭代次数影响下的时延性能比较

Fig. 2　Comparison of delay performance under the influence of the number of iterations

图 3 仿真对比了在 Zipf 分布影响因子下 ACSAM、MAP、DLC 及 RC 的缓存性能。假定服务内容数为 1000，MEC 和车辆数分别为 10 和 20，从中可以看出，除 RC 缓存策略外，其他三种缓存策略的延迟都随着影响因子的增大而降低。这是由于在 RC 中，每个缓存节点随机地确定其缓存项，不受流行度或其他参数影响，性能相当不稳定，而其他三种策略受流行度影响，当被重复请求的内容较少时，请求内容在车辆、MEC 服务器等接近用户边缘的地方缓存的概率增加，从而提高了缓存性能，降低了时间消耗。综上，随着 Zipf 分布影响因子的变化，ACSAM 的缓存策略在降低时延性能方面优于其他缓存策略。

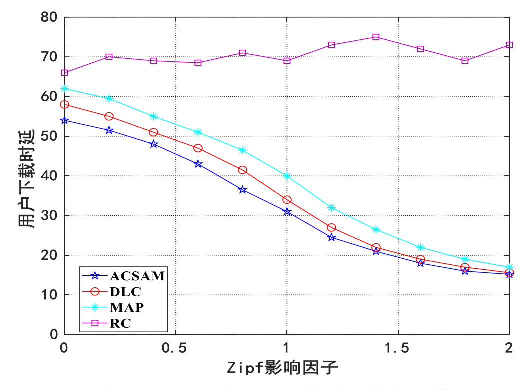


图 3　Zipf 影响因子下的时延性能比较

Fig. 3　Comparison of delay performance under Zipf impact factor

图 4、5 描述了在 MEC 服务器数量影响下 ACSAM、MAP、DLC 和 RC 四种缓存方案的性能差异。假定有 1000 个服务内容，并且内容流行度服从影响因子为 0.56 的 Zipf 分布。从图 4、5 中可以看出，随着 MEC 设备数量的增多，四种缓存策略的缓存命中率皆升高，用户平均下载时延都降低。这是因为越来越多的请求内容可以直接从多个 MEC 协作的缓存域中获取，从而在提高缓存命中率的同时减少了从远程云下载的传输冗余，并且由于 ACSAM 算法能有效避免局部最优，因此相比于其他三种缓存方案可保证内容被缓存在最佳的节点，故而有着更高的缓存命中率和更低的下载时延。具体地，当 MEC 服务器数量为 30 时，ACSAM 缓存策略的缓存命中率比 DLC、 MAP 和 RC 分别高约 7.9%、18.9%、27.8%，在时延上分别低约 38.9%、47.4%、76.2%。

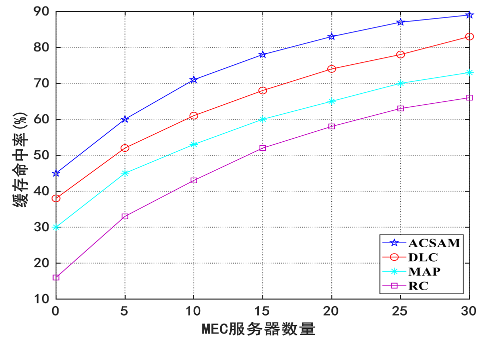


图 4　MEC 服务器数量变化下的缓存命中率比较

Fig. 4　Comparison of cache hit rates under changes in the number of MECS

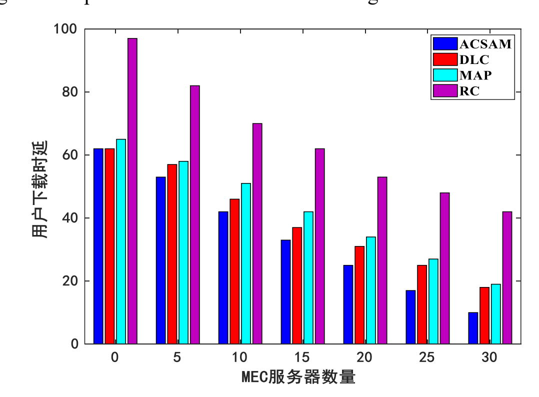


图 5　MEC 服务器数量变化下的时延性能比较

Fig. 5　Comparison of delay performance under changes in the number of MECS

图 6、7 仿真对比了当服务内容数量不同时，ACSAM、MAP、DLC 和 RC 四种缓存策略的性能。假定 MEC 的数量为 10，车辆数为 20，Zipf 分布影响因子的值为 0.56。由图 6、

7 可以看出，随着服务内容数量的增加，所有缓存策略的缓存命中率均保持下降趋势，这是因为车辆和 MEC 服务器的存储空间都是有限的，当发布的内容数量累积到一定程度时，仅依靠车辆、MEC 无法满足用户的需求，越来越多的内容需要从核心网进行下载。并且通过仿真结果可得，ACSAM 缓存策略要优于 DLC、MAP 和 RC。具体地，当服务内容数量为 1000 时，ACSAM 缓存策略的缓存命中率比 DLC、MAP 和 RC 分别高出了大约 9.4%、22.5%、36.3%，平均下载时延比 DLC、MAP 和 RC 分别低约 10.9%、14.6%、40.6%。

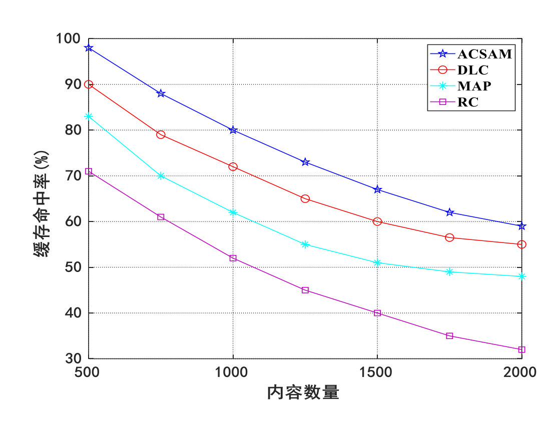


图 6 服务内容数量变化下的缓存命中率比较

Fig. 6 Comparison of cache hit ratios with changes in service content

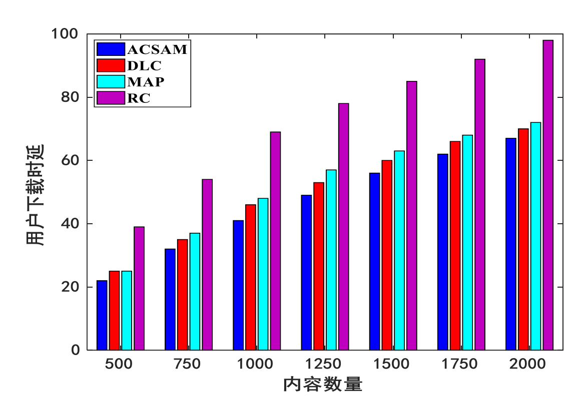


图 7 服务内容数量变化下的时延性能比较

Fig. 7 Comparison of latency performance with changes in service content

5 结束语

车联网面临着海量数据流量的冲击，为了实现车联网业务的实时响应，本文在基于 5G 的“车-边-云”协同系统架构下，建立了基于时延的通信模型，实现了对车、MEC 设备、云三方面资源的调度，同时提出了 ACSAM 缓存策略，通过该策略可获得最小内容下载时延，从而保证各内容被缓存在最佳的节点。仿真结果证明，在“车-边-云”协同系统架构下，ACSAM 缓存策略能有效减少内容的下载时延，提高响应速度。但由于本文仅假设了车辆资源共享的理想情况，不符合实际情形，因此，后续的研究会将车的自利性考虑在内，通过设立奖励机制实现车联网中各参与者的利益最大化。

参考文献：

[1] Cheng X, Zhang R, Chen S, *et al.* 5G enabled vehicular communications and networking [J]. China Communications, 2018, 15 (7): 3-6.

[2] Bastug E, Bennis M, Debbah, Mérouane. Living on the edge: The role of proactive caching in 5G wireless networks [J]. IEEE Communications Magazine, 2014, 52 (8): 82-89.

[3] An Liu, Vincent K. N. Lau. Cache-Enabled Opportunistic Cooperative MIMO for Video Streaming in Wireless Systems [J]. IEEE Transactions on Signal Processing: A publication of the IEEE Signal Processing Society, 2014, 62 (2): 390-402.

[4] 左超，武继刚，史雯隽. 移动边缘计算中的端到端任务分配算法 [J]. 计算机应用研究, 2019, 37 (7): 1001-3695. (Zuo Chao, Wu Jigang, Shi Wenji. End-to-end task assignment algorithm in mobile edge computing [J]. Journal of Computer Applications, 2019, 37 (7): 1001-3695.)

[5] Jiang L, Feng G, Qin S. Cooperative content distribution for 5G systems based on distributed cloud service network [C]// IEEE International Conference on Communication Workshop, 2015: 1125-1130.

[6] Tran T X, Pompili D. Adaptive Bitrate Video Caching and Processing in Mobile-Edge Computing Networks [J]. IEEE Transactions on Mobile Computing, 2019, 18 (9): 1965-1978.

[7] Thar K, Tran N H, Oo T Z, *et al.* DeepMEC: Mobile Edge Caching Using Deep Learning [J]. IEEE Access, 2018: 78260-78275.

[8] Mahmood A, Casetti C, Chiasserini C, *et al.* Mobility-aware edge caching for connected cars [C]. wireless on demand network systems and service, 2016: 1-8.

[9] Ma J, Wang J, Liu G, *et al.* Low Latency Caching Placement Policy for Cloud-Based VANET with Both Vehicle Caches and RSU Caches [C]. global communications conference, 2017: 1-6.

[10] 徐灿辉. 5G 网络架构对车联网发展影响的研究 [J]. 广东通信技术, 2018, 38 (01): 76-79. (Xu Canhui. Research on the Influence of 5G Network Architecture on the Development of Telematics [J]. Guangdong Communication Technology, 2018, 38 (01): 76-79.)

[11] Zheng K, Liu F, Zheng Q, *et al.* A graph-based cooperative scheduling scheme for vehicular networks [J]. IEEE transactions on vehicular technology, 2013, 62 (4): 1450-1458.

[12] Luoto P, Bennis M, Pirinen P, *et al.* Vehicle clustering for improving enhanced LTE-V2X network performance [C]// European Conference on Networks & Communications. IEEE, 2017: 1-5.

[13] Tao M, Chen E, Zhou H, *et al.* Content-Centric Sparse Multicast Beamforming for Cache-Enabled Cloud RAN [J]. IEEE Transactions on Wireless Communications, 2016, 15 (9): 6118-6131.

[14] 侯停停. 移动边缘计算 (MEC) 架构中高效内容分发技术 [D]. 四川: 电子科技大学, 2018. (Hou Tingting. Efficient content distribution technology in mobile edge computing (MEC) architecture [D]. Sichuan: University of Electronic Science and technology, 2018.)

[15] Saichanma S, Chulsomlee S, Thangrua N, *et al.* Exact algorithms for the joint object placement and request routing problem in content distribution networks [J]. Computers & Operations Research, 2008, 35 (12): 3860-3884.